**INSTITUTUL DE CERCETARE ŞTIINŢIFICĂ
ŞI INGINERIE TEHNOLOGICĂ
PENTRU TEHNICA DE CALCUL ŞI INFORMATICĂ
SECTOR TEHNICĂ DE CALCUL**

COBRA BASIC

MANUAL DE UTILIZARE

BRAŞOV

1988

Editat de I.T.C.I. Braşov

Bun de tipar: 12.04.1988

Tiparul executat sub comanda nr. 1239
Întreprinderea Poligrafică Braşov

1. INTRODUCERE

Acest manual prezintă limbajul BASIC pentru microcalculatorul COBRA.

Numele COBRA, este o pură coincidenţă cu cel al reptilei indiene „NAJA”; el a rezultat din asocierea cuvintelor COmputer BRAşov.

COBRA este un microcalculator pe 8 biţi, din clasa calculatoarelor personal-profesionale, realizat în cadrul Filialei Braşov a I.C.S.I.T.-T.C.I. (Institutul de Tehnică de Calcul şi Informatică).

Funcţie de opţiunea utilizatorului, calculatorul se poate configura în maşina BASIC-SPECTRUM (100% compatibil din punct de vedere al softului cu calculatorul SINCLAIR ZX SPECTRUM) sau în maşina CP/M (detaliile privind organizarea şi funcţionarea sistemului în acest regim sînt date în manualul COBRA CP/M).

Calculatorul este format din următoarele blocuri:

— unitate centrală cu microprocesor Z80, la frecvenţa de 3,5 MHz;

— memorie EPROM de 16 Kocteţi ce conţine interpretorul BASIC;

— memoria RAM de 64 Kocteţi;

— controler video care permite afişarea pe monitor alb-negru, cu 8 nivele de gri sau pe monitor color cu 8 culori şi o rezoluţie de 194×256 de pixeli;

— porturi de intrare/ieşire ce permit conectarea tastaturii, casetofonului (pentru salvarea şi încărcarea programelor de pe casete magnetice), difuzorului (semnalizări sonore), interfeţei seriale RS232 (pentru cuplarea imprimantei etc.), unui joystick Kempston şi o serie de linii disponibile pentru alte interfaţări — interfaţa pentru unităţi de memorie externă cu discuri flexibile de 5¼” şi/sau 8”.

2.1. Tastatura

Prin intermediul tastaturii se pot introduce fie simboluri (litere, cifre etc.), fie comenzi compuse (nume de funcţii, cuvinte cheie etc.). Diferitele funcţii ale aceleiaşi taste se obţin prin apăsarea simultană a unei taste de shiftare (CAPS SHIFT sau SYMBOL SHIFT) împreună cu tasta respectivă şi avînd calculatorul în diferite moduri de lucru. Modul de lucru este specificat de cursor, o literă clipind, care indică semnificaţia ce va fi atribuită tastelor, după cum urmează:

K („Keywords” = cuvinte cheie).

Calculatorul trece automat din modul L în modul K, atunci cînd trebuie să urmeze introducerea unei comenzi (printr-un cuvînt cheie) sau a unei linii noi de program (printr-un număr). Aceasta se poate întîmpla la începutul unei linii sau după comanda THEN sau după două puncte (:) (mai puţin în interiorul unui string = şir de caractere alfanumerice). În modul K, o tastă neshiftată va fi interpretată ca şi un cuvînt cheie (comandă) sau o cifră.

L („Letters” = litere mici).

În oricare altă situaţie, calculatorul intră în modul L. Neshiftată, o tastă va fi interpretată ca literă (mică) sau cifră (simbolul principal al tastei respective).

C („Capitals” = litere mari).

Este o variantă a modului L, în care literele sînt interpretate drept litere mari.

Se poate trece din modul L în modul C şi invers tastînd CAPS LOCK (CAPS SHIFT şi 2).

În continuare sînt prezentate diferitele interpretări ale tastelor, funcţie de modul de lucru şi tastele de shiftare:

Modul de lucru	Tasta de shiftare	Tasta	Interpretare
K	fără	literă	comanda de jos (neagră)
K	SYMBOL SHIFT	literă	comanda de sus (roşie)
K	fără	cifră	cifră
K	SYMBOL SHIFT	cifră	simbolul (roşu) de pe tastă
L	fără	literă	literă mică
L	SYMBOL SHIFT	literă	simbolul (comanda) roşu de pe tastă de sus
L	CAPS SHIFT	literă	literă mare
L	fără	cifră	cifră
L	SYMBOL SHIFT	cifră	simbolul (roşu) de pe tastă

E („Extended” = extins).

Acest mod este utilizat pentru obţinerea celorlalte comenzi, din josul tastelor. Se intră în modul E apăsînd tasta EXTENDED MODE şi se menţine acest mod pînă după prima (oricare) tastare.

În modul E, o tastă neshiftată va fi interpretată ca şi comanda de deasupra ei, shiftată (cu SYMBOL SHIFT sau CAPS SHIFT) va fi interpretată ca şi comanda din josul tastei.

G („Graphics” = grafic).

Se intră tastînd GRAPHICS (CAPS SHIFT simultan cu cifra 9) şi se menţine acest mod pînă la repetarea acestei comenzi. În acest mod cifrele (cu excepţia lui 9 şi 0) vor da simbolurile grafice desenate pe taste iar literele (cu excepţia lui V, W, X, Y şi Z) vor da simboluri grafice definite de utilizator.

Orice tastă menţinută apăsată mai mult de 2—3 secunde va fi repetată.

Intrările de la tastatură apar pe ecranul monitorului TV, în partea de jos, înaintea cursorului.

Cursorul poate fi mutat în stînga apăsînd tasta ← (CAPS SHIFT şi 5) sau dreapta apăsînd tasta → (CAPS SHIFT şi 8).

Caracterul dinaintea cursorului poate fi şters cu DEL (CAPS SHIFT şi 0).

O linie întreagă poate fi ştearsă tastînd EDIT (CAPS SHIFT şi 1) urmată de ENTER.

După tastarea lui ENTER o linie este fie executată, fie introdusă într-un program, fie folosită ca date de intrare (într-o instrucţiune INPUT), dacă nu conţine erori de sintaxă. Cînd linia conţine erori, după eroare apare un semn de întrebare (?) clipind.

Liniile de program deja introduse sînt afişate în partea de sus a ecranului. Ultima linie introdusă (linia curentă) este indicată de simbolul cursor de linie „>". Acesta poate fi deplasat în sus cu tasta ↑ (CAPS SHIFT şi 7) sau în jos cu tasta ↓ (CAPS SHIFT şi 6), la alte linii. Linia curentă poate fi editată, în partea de jos a ecranului, cu EDIT (CAPS SHIFT şi 1).

În timpul execuţiei unei comenzi sau a unui program mesajele sînt afişate în partea de sus a ecranului, unde rămîn pînă ce este introdusă o linie de program sau se tastează ENTER după o linie goală.

În partea de jos a ecranului pot să apară mesaje (litere sau cifre), în situaţii normale sau în cazul unor erori. Mesajele conţin şi numărul liniei pe care se află ultima instrucţiune executată şi un număr indicînd poziţia acelei instrucţiuni în cadrul liniei respective.

În anumite situaţii CAPS SHIFT împreună cu SPACE acţionează ca o întrerupere (BREAK), oprind executarea unui program.

2.2. Ecranul monitorului TV

Ecranul conţine 24 de linii, fiecare de cîte 32 de caractere.

În partea de sus a ecranului (22 de linii) se pot afişa liniile unui program sau diferite mesaje, liniile completîndu-se de sus în jos. Cînd toate cele 22 de linii s-au umplut, se face automat o defilare („SCROLL"), toate liniile fiind mutate cu o poziţie mai sus. După defilarea unei pagini, calculatorul aşteaptă apăsarea oricărei taste pentru a continua cu următoarea pagină; tastele N, SPACE (BREAK) sau STOP opresc defilarea şi se aşteaptă o comandă. Dacă se tastează N sau SPACE sau STOP programul se întrerupe şi apare mesajul „D BREAK-CONT repeats".

Partea de jos a ecranului (2 linii) este utilizată pentru introducerea comenzilor, a liniilor noi de program, pentru introducerea datelor, pentru editarea liniilor de program sau pentru afişarea unor mesaje trimise de calculator.

Fiecare poziţie a unui caracter are o serie de atribute care stabilesc culoarea fondului (hîrtia=paper) şi a caracterelor (cerneala=ink), strălucirea (normală sau mărită) şi modul continuu sau intermitent (pîlpîind) de afişare pe ecran.

Culoarea pentru marginea ecranului se stabileşte cu instrucţiunea (declaraţie) BORDER n (n=codul asociat culorii, ca şi în instrucţiunile PAPER, INK, prezentate mai jos).

O poziţie pentru un caracter este alcătuită din 8 × 8 pixeli. Se poate obţine o rezoluţie grafică bună, stabilind caracterul fiecărui pixel (ink sau paper), individual.

Atributele fiecărui caracter şi/sau pixel se stabilesc prin valorile ce se atribuie „parametrilor de tipărire": PAPER, INK, FLASH, BRIGHT,

INVERSE şi OVER. Există două seturi de parametri de tipărire: permanenţi şi temporari.

Cînd PAPER, INK, ... sînt utilizaţi ca şi instrucţiuni BASIC independente, atributele astfel stabilite sînt parametrii permanenţi, valorile lor rămînînd valabile pînă la redefinirea lor.

Valorile iniţiale pentru partea superioară a ecranului sînt: caractere negre pe fond alb, strălucire normală, fără pîlpîire, video normal, fără supraimprimare.

Valorile iniţiale pentru partea de jos a ecranului sînt: fondul are aceeaşi culoare cu borderul (marginea ecranului), caracterele sînt negre sau albe, strălucirea normală, fără pîlpîire, video normal, fără supraimprimare.

Cînd PAPER, INK, ... sînt folosiţi într-o altă instrucţiune BASIC, cum ar fi PRINT, INPUT, PLOT, DRAW, atributele stabilite au caracter temporar, fiind valabile doar pe perioada execuţiei instrucţiunii respective (PRINT, INPUT, ...), după care sînt înlocuite cu valorile date de parametrii de tipărire permanenţi.

Parametrilor BORDER, PAPER şi INK li se pot atribui valori între 0 şi 9. Valorile între 0 şi 7 corespund culorilor:

0 — negru;
1 — albastru;
2 — roşu;
3 — magenta (violet);
4 — verde;
5 — cyan (turcoaz);
6 — galben;
7 — alb.

Valoarea 8 („transparentă") va lăsa culoarea ecranului neschimbată (se afişează un caracter cu aceeaşi culoare cu cea a fondului).

Valoarea 9 („contrast") produce tipărirea cu negru sau alb, pentru obţinerea unui contrast cît mai mare în raport cu celelalte culori.

Parametrilor FLASH şi BRIGHT li se pot atribui valorile 0, 1 sau 8. 0 produce afişarea normală, 1 produce pîlpîirea, respectiv strălucirea mărită a caracterului, 8 („transparentă") nu produce nici o schimbare.

Parametrilor OVER şi INVERSE li se pot atribui valorile 0 sau 1:

OVER 0 — vechile caractere afişate sînt şterse şi în locul lor sînt afişate noile caractere.

OVER 1 — tipărirea noilor caractere se face peste cele vechi, acestea fiind păstrate (supraimprimare); se pot afişa astfel caractere compuse, cum ar fi literele cu accente, trema etc.

INVERSE 0 — (video normal) caracterele sînt de culoarea cernelii pe fond de culoarea hîrtiei.

INVERSE 1 — (video invers) se afişează caractere de culoarea hîrtiei pe fond de culoarea cernelii

Caracterul de control TAB n produce tipărirea caracterului în coloana q unde q este restul împărţirii lui n la 32 (modulo 32).

Caracterul de control virgula produce tipărirea unui număr suficient de pauze (cel puţin una) pentru ca tipărirea să se facă în coloana 0 sau 16.

Caracterul de control ENTER produce tipărirea pe linia următoare.

3. LIMBAJUL BASIC

În acest capitol se prezintă particularităţile implementării limbajului BASIC pe microcalculatorul COBRA.

Numerele sînt reprezentate extern cu o precizie de 9 sau 10 cifre. Cel mai mare număr acceptat este aprox. 1*E+38 (10 ridicat la puterea 38) şi cel mai mic (număr pozitiv) este aprox. 4*E-39 (4*10 la puterea -39).

Numerele sînt memorate intern în formă binară, virgulă mobilă, cu un octet pentru exponentul e (1 <= e <= 225) şi 4 octeţi pentru mantisa m (1/2 <= m < 1). Deoarece 1/2 <= m < 1, bitul cel mai semnificativ al mantisei este totdeauna 1. Aceasta permite înlocuirea lui cu bitul de semn: 0 pentru numere pozitive şi 1 pentru numere negative.

Numerele întregi au o reprezentare specială în care primul octet este 0, al doilea este octetul de semn (0 sau FFh), octeţii 3 şi 4 reprezentînd valoarea sub forma complementului de ordinul 2, cu octetul cel mai puţin semnificativ pe primul loc.

Variabilele numerice simple pot avea nume de lungime arbitrară, începînd obligatoriu cu o literă urmată sau nu de alte litere sau cifre. Spaţiile sînt ignorate. Toate literele sînt transformate în litere mici.

Variabilele de control în cicluri FOR-NEXT trebuie să aibă numele format dintr-o singură literă.

Variabilele numerice indexate (matriciale) trebuie să aibă numele format dintr-o singură literă (care poate să şi coincidă cu numele unei variabile numerice simple), urmată de o paranteză în care sînt specificaţi indicii. Aceştia pot să aibă orice valoare (pozitivă începînd de la 1) şi pot fi oricîţi (dimensiuni arbitrare).

Variabilele şir simple („Strings”) pot avea lungimea arbitrară. Numele trebuie să fie format dintr-o singură literă, urmată de semnul dolar $. Valoarea atribuită variabilei trebuie închisă între ghilimele şi este constituită dintr-un şir de caractere alfanumerice.

Variabilele şir indexate („String arrays”) pot avea dimensiuni arbitrare (număr arbitrar de indici), de mărimi arbitrare. Numele este format dintr-o singură literă (care trebuie să fie diferită de numele oricărei variabile şir simple) urmată de dolar $ şi de paranteză conţinînd numele indicilor. Toate şirurile ce compun o variabilă şir indexată au aceeaşi lungime, stabilită printr-o instrucţiune de tip DIM. Valorile indicilor sînt pozitive, începînd cu 1.

3.1 Funcţiile BASIC

Argumentul unei funcţii nu trebuie pus între paranteze, dacă este o constantă sau o variabilă.

Funcţia	**Tipul argumentului: (x)**	**Rezultatul**
ABS	Număr	Valoarea absolută
ACS	Număr	Arccos în radiani. Eroare A dacă x este în afara intervalului [-1,+1].
AND	Operandul drept totdeauna un număr	Operaţia binară ŞI
	Operandul stîng este un număr	A AND B = A dacă B<>0 A AND B = 0 dacă B=0
	Operandul stîng este un string	A$ AND B = A$ dacă B<>0 A$ AND B = A$ dacă B=0
ASN	Număr	Arcsin în radiani. Eroare A dacă x nu este în intervalul [-1,+1].
ATN	Număr	Arctg în radiani
ATTR	Două argumente numerice x şi y, închise în paranteze	Un număr al cărui cod binar reprezintă atributele caracterului de pe linia x, coloana y, pe ecran. Bitul 7 (cel mai semnificativ) este 1 pentru clipire, 0 pentru continuu. Bitul 6 este 1 pentru strălucire, 0 pentru normal. Biţii 5-3 formează culoarea hîrtiei (paper). Biţii 2-0 formează culoarea cernelii (ink). Eroare B dacă x şi y nu sînt în intervalele 0 <= x <= 23 respectiv 0 <= y <= 31.
BIN	Număr în binar	Aceasta nu este o funcţie propriu-zisă, ci o notaţie alternativă pentru numere. BIN urmat de o secvenţă de 0 şi 1 transformă această secvenţă în numărul corespondent în baza 10.
CHR$	Număr	Aproximează x cu cel mai apropiat întreg şi dă caracterul ASCII al acelui întreg.
CODE	String	Codul primului caracter din secvenţă ce formează stringul (sau 0 dacă stringul nu conţine nici un caracter).
COS	Număr (în radiani)	Cos x
EXP	Număr	Exp x

Funcţia	Tipul argumentului: (x)	Rezultatul
FN	F(x, y, ...)	FN urmată de o literă apelează o funcţie definită-utilizator prin DEF FN. Argumentele funcţiei trebuie să fie închise între paranteze, chiar în absenţa argumentelor, parantezele trebuie puse după numele funcţiei: FN F().
IN	Număr	Procesorul citeşte valoarea din portul x (0 <= x <= FFFFh) (se încarcă valoarea x în perechea de regiştri bc şi se efectuează instrucţiunea IN A(C) din limbajul de asamblare Z80).
INKEY$	Fără	Se citeşte instantaneu tastatura. Rezultatul este caracterul, reprezentînd (în modurile L sau C) tasta apăsată sau stringul gol: ””
INT	Număr	Partea întreagă a numărului (rotunjit totdeauna la valoarea mai mică).
LEN	String	Numărul caracterelor ce alcătuiesc stringul.
LN	Număr	ln x (logaritm natural, în baza e). Eroare A dacă x <= 0.
NOT	Număr	0 dacă x <> 0 1 dacă x = 0 NOT are prioritatea 4
OR	Operaţia binară SAU. Ambii operanzi sînt numere.	a OR b = 1 dacă b <> 0 a OR b = a dacă b = 0 OR are prioritatea 2
PEEK	Număr	Conţinutul adresei x a memoriei (cu x rotunjit la cel mai apropiat întreg). Eroare B dacă x nu este în intervalul [0, 65535] (64 kocteţi)
PI	Fără	PI (3.14159265...)
POINT	Două argumente x, y, ambele numere, cuprinse între paranteze.	1 dacă pixelul de coordonate x, y are culoarea cernelii (ink). 0 dacă pixelul de coordonate x, y are culoarea hîrtiei (paper). Eroare B dacă x şi y nu sînt în intervalele 0 <= x <= 255 respectiv 0 <= y <= 175.
RND	Fără	Generarea unui număr aleator y cuprins în intervalul 0 <= y <= 1
SCREEN$	Două argumente x, y, ambele numere, cuprinse între paranteze.	Caracterul ce apare pe ecranul TV în linia x, coloana y. Dacă caracterul nu este recunoscut, rezultă stringul gol (””). Eroare B dacă x şi y nu sînt în intervalele 0 <= x <= 23 respectiv 0 <= y <= 31.

Funcţia	Tipul argumentului: (x)	Rezultatul
SGN	Număr	Semnul lui x: -1 pentru x < 0 0 pentru x = 0 +1 pentru x > 0
SIN	Număr (în radiani)	Sin x
SQR	Număr	Radical din x. Eroare A pentru x < 0.
STR$	Număr	Caracterul ASCII alfanumeric al lui x.
TAN	Număr (în radiani)	Tg x (tangenta x).
USR	Număr	Apelarea unei subrutine în cod maşină a cărei adresă de început este x. La revenire, rezultatul este conţinutul registrului pereche bc.
USR	String	Adresa şirului de biţi ai caracterului grafic definit de utilizator, corespunzător stringului x. Eroare A dacă x nu este o literă unică între a şi u sau un caracter grafic definit de utilizator.
VAL	String	Evaluarea numerică a lui x (mai puţin ghilimelele care-l cuprind). Eroare C dacă x conţine erori de sintaxă.
VAL$	String	Evaluarea lui x (mai puţin ghilimelele) ca expresie alfanumerică (string). Eroare C dacă x conţine erori de sintaxă.
—	Număr	Înmulţire cu -1 (schimbarea semnului).

3.1 Operaţii binare

+	Adunare (numere sau stringuri)	
−	Scădere	
×	Înmulţire	
/	Împărţire	
^	Ridicare la putere. Eroare B dacă operandul stîng este negativ.	
=	Egalitate	Ambii operanzi trebuie să fie de
>	Mai mare ca	acelaşi tip.
<	Mai mic ca	Rezultatul este:
<=	Mai mic cel mult egal	
>=	Mai mare cel puţin egal	1 dacă condiţia este adevărată
<>	Inegalitate	0 în caz contrar

3.3 Priorităţile funcţiilor şi operaţiilor

Operaţia	Prioritatea
Indexarea termenilor unui şir şi extragerea unui subşir dintr-un şir (slicing)	**12**
Toate funcţiile cu excepţia lui NOT şi a schimbării semnului	**11**
^	**10**
Schimbarea semnului	**9**
×, /	**8**
+, -	**6**
=, >, <, <=, >=, <>	**5**
NOT	**4**
AND	**3**
OR	**2**

3.4 Instrucţiuni BASIC

În lista ce urmează se folosesc notaţiile:
a = literă
v = variabilă
x, y, z= expresie numerică
m, n = expresie numerică rotunjită la întregul cel mai apropiat
e = expresie
f = string evaluat
s = secvenţa de instrucţiuni separate prin două puncte (:)
c = secvenţa de caracteristici de culoare, cu separatori virgulă sau punct-virgulă, de forma: PAPER, INK, FLASH, BRIGHT, INVERSE, OVER.

Sînt permise expresii arbitrare în orice poziţie, cu excepţia numerelor cu care trebuie să înceapă liniile de program.

Toate instrucţiunile, exceptînd DEF FN şi DATA pot fi utilizate în programe, sau ca şi comenzi direct executabile.

O linie program sau o comandă poate să conţină una sau mai multe instrucţiuni separate prin două puncte (:). Cu excepţia lui IF şi REM nu există restricţii cu privire la locul pe care îl poate ocupa o anumită instrucţiune în cadrul unei linii program.

BEEP x, y

Se emite un sunet cu durata de x secunde şi înălţimea cu y semitonuri deasupra (sau sub cînd y < 0) notei „la” din octava a 4-a (440 Hz).

Exemplu: Folosind instrucţiunea BEEP 1, 2 se emite timp de o secundă nota „si” din octava a 4-a.

BORDER m

Stabileşte culoarea marginii ecranului (border) şi a cernelii în partea de jos a ecranului. Eroare K dacă m nu este în intervalul [0, 7].

BRIGHT n

Stabileşte culoarea caracterelor ce vor fi afişate:

n = 0 strălucire normală;
n = 1 strălucire mărită;
n = 8 transparenţă.

Eroare K dacă n diferit de 0, 1, 8.

CAT

Vezi MON.

CIRCLE x, y, z

Desenează pe ecran un arc de cerc cu centrul în punctul (x, y) şi de rază z.

Exemplu: Dacă dorim trasarea unui cerc avînd centrul în punctul de coordonate (128, 120) şi de rază 50, vom folosi instrucţiunea:

CIRCLE 128, 120, 50

CLEAR

Şterge toate variabilele, eliberînd memoria pe care o ocupau, se execută RESTORE, CLS, se resetează poziţia de PLOT (se poziţionează cursorul grafic în punctul stînga-jos al ecranului) şi se goleşte stiva pentru GO SUB.

Nu modifică valoarea variabilei de sistem RAMTOP.

CLEAR n

Similar cu CLEAR, în plus se atribuie valoarea n variabilei de sistem RAMTOP, iar stiva pentru GO SUB începe de la noul RAMTOP.

Exemplu: Dacă dorim ca noul RAMTOP să fie la adresa 32500, vom scrie instrucţiunea: **CLEAR 32500**.

CLOSE $

Neutilizabil.

CLS

(Clear Screen) Şterge ecranul TV.

CONTINUE

CONT pe tastatură. Reluarea unui program din poziţia unde rularea a fost întreruptă, cu mesaj de întrerupere diferit de 0. Dacă mesajul a fost 9 sau L, reluarea se face cu următoarea instrucţiune (incluzîndu-se şi instrucţiunile de salt); în celelalte cazuri reluarea se face inclusiv cu instrucţiunea unde s-a produs întreruperea. Dacă mesajul de întrerupere apare într-o linie de comandă atunci CONTINUE va încerca reluarea din acea linie de comandă şi se va intra într-o buclă dacă eroarea a fost 0:1 sau se va afişa mesajul 0 dacă eroarea a fost 0:2 sau se va afişa mesajul N dacă eroarea a fost 0:3 sau mai mare.

COPY

Se obţine o copie a ecranului (24 linii), cu 8 nivele de gri, la o imprimantă grafică de tip RCD 9335.

Listarea poate fi întreruptă prin apăsarea tastei BREAK.

DATA e1, e2, e3

Porţiune dintr-o listă DATA (cuplată cu instrucţiunea READ).

Exemplu: după executarea secvenţei de program de mai jos

```
10 READ A, B, C;
15 DATA 10, 15, 35
```

variabilele A, B, C, vor avea valorile 10, 15, respectiv 35.

DEF FN f(a1,...,aK)=e

Definirea unei funcţii-utilizator; trebuie să se găsească într-un program. f, a1, ... aK sînt litere sau litere urmate de $ pentru argumente şi rezultate sub formă de stringuri. În cazul în care nu există argumente se va utiliza forma DEF FN f()=e.

Exemplu: Fie secvenţa de program de mai jos:

```
10 LET x=10: LET y=5: LET a=50;
15 DEF FN f(x, y)=a+x*y
20 DEF FN g()=a+x*y
25 LET p=FN f(7, 8): LET r=FN g()
```

După executarea acestei secvenţe de program vom avea p=50+7*8 şi r=50+10*5, adică p=106 şi r=100, adică în calculul lui p se iau argumentele cu valori 7 şi 0 definite în linia 25, iar în calculul lui r rămîn argumentele definite în linia 10.

DELETE f

Neutilizabil.

DIM a(n1,...,nK)

Şterge toate vechile matrici cu numele a şi deschide o matrice numerică a avînd K dimensiuni n1, ... nK. Iniţializează toate elementele matricii a atribuindu-le valoarea 0.

Exemplu: Dacă vrem să folosim matricea A avînd 3 linii şi 4 coloane o vom defini prin instrucţiunea:

```
10 DIM A(3,4)
```

DIM a$(n1,...,nK)

Şterge orice matrice de stringuri sau string avînd numele a$ şi deschide o matrice de caractere cu K dimensiuni n1, ... nK. Iniţializează toate elementele atribuindu-le valoarea stringului gol ””. Eroare 4 dacă se depăşeşte memoria RAM. Matricile nu sînt complet definite pînă cînd nu sînt dimensionate printr-o instrucţiune de tip DIM.

DRAW x, y, z

Desenează pe ecran un arc de cerc pornind din poziţia actuală a cursorului grafic (ultimul punct desenat pe ecran), şi pînă în punctul depărtat cu x pixeli pe orizontală şi y pixeli pe verticală, arc de cerc avînd mărimea de z radiani.

DRAW x, y

Echivalent cu DRAW x, y, 0. Trasarea unei drepte pornind din poziţia actuală a cursorului grafic pînă în punctul depărtat cu x pixeli pe orizontală şi y pixeli pe verticală. Eroare B dacă se iese de pe ecran.

Exemplu: Dacă poziţia cursorului este în punctul (100, 100) şi dorim să trasăm o dreaptă pînă în punctul (120, 145) vom folosi instrucţiunea:

```
10 DRAW 20,45
```

ERASE

Neutilizabil.

FLASH n

Stabileşte modul staţionar sau clipind în care să se afişeze un caracter pe ecranul TV.

n = 0 staţionar;
n = 1 clipind;
n = 8 nu se face nici o schimbare.

FOR a=x TO y STEP z

Intrarea într-un ciclu indexat de variabila a căreia i se atribuie valori între x şi y cu pasul z. Sfîrşitul instrucţiunilor ce formează ciclul este indicat printr-o instrucţiune NEXT a.

FOR a=x TO y

La fel ca mai sus dar cu pasul avînd valoarea implicită 1.

Exemplul 1:
Fie secvenţa de mai jos:

```
10 FOR n=1 TO 19 STEP 2
......................................
50 NEXT n
```

Se vor executa toate liniile de program avînd numerele de secvenţă între 10 şi 60 pentru n=1,3,5,7,9,11,13,15,17,19, deci se va parcurge secvenţa de instrucţiuni cu numerele de ordine între 10 şi 50 de 10 ori. La ieşirea din corpul ciclului variabila n va avea valoarea 21.

Exemplul 2:
Fie secvenţa de program de mai jos:

```
10 FOR m=1 TO 9
......................................
60 NEXT m
```

Se vor executa toate liniile de program avînd numerele de secvenţă între 10 şi 60 pentru m=1,2,3,4,5,6,7,8,9, deci se va parcurge secvenţa de instrucţiuni cu numerele de ordine între 10 şi 60 de 9 ori. La ieşirea din corpul ciclului variabila m va avea valoarea 10.

FORMAT f

Neutilizabil.

GOSUB n

Se memorează numărul liniei pe care se află instrucţiunea GOSUB n în stivă, apoi se execută un salt la linia n. Eroare 4 dacă nu sînt suficiente instrucţiuni de revenire RETURN.

GOTO n

Salt la linia n. Dacă aceasta nu există, saltul se face la prima linie întîlnită după n.

IF x THEN s

Dacă condiţia x este adevărată (rezultatul diferit de zero) se execută s (s semnifică toate instrucţiunile pînă la sfîrşitul liniei; s nu poate fi un număr de linie).

Exemplu de folosire a instrucţiunilor GOTO, GO SUB şi IF.
Fie programul de mai jos:

```
10 DIM a(5)
20 FOR i=1 TO 5: READ a(i): NEXT i;
30 DATA 6,3,4,2,8;
40 FOR n=1 TO 5;
50 GO SUB 200;
60 PRINT ”VALOAREA LA IEŞIRE ESTE:”; w;
70 NEXT n;
80 STOP;
200 IF a(n)>10 THEN GO TO 230;
210 LET w=a(n)+10;
220 GO TO 240;
230 LET w=a(n)+20;
240 RETURN.
```

Se vor citi valorile a(1)=6, a(2)=3, a(3)=4, a(4)=2, a(5)=8, apoi pentru fiecare valoare a vectorului ”a” se va parcurge secvenţa de instrucţiuni de la 200 la 240, calculîndu-se de fiecare dată valoarea variabilei w, după care se va face o revenire la eticheta 60 şi se va afişa valoarea lui w. La terminarea acestui program vom avea pentru w pe rînd valorile 16, 13, 14, 12.

INK n

Stabileşte culoarea cu care vor fi afişate următoarele caractere: n cuprins în intervalul [0, 7] pentru o culoare, n=8 pentru transparenţă, n=9 pentru contrast. Eroare K dacă n nu este în intervalul [0, 9].

INPUT ”...”

”...” este o secvenţă de caractere separate ca şi în instrucţiunea PRINT prin virgule, punct-virgule sau apostroafe. Caracterele pot fi:

i) Orice caracter tipăribil (acceptat într-o instrucţiune de tip PRINT), care să nu înceapă cu o literă.
ii) Numele unei variabile.
iii) Instrucţiunea LINE urmată de numele unei variabile de tip string (şir de caractere alfanumerice).

În continuare sînt date detalii pentru fiecare din cele trei cazuri:

i) Caracterele tipăribile şi elementele de separare (virgule, punct-virgule, apostroafe) sînt tratate exact ca într-o instrucţiune PRINT, exceptînd faptul că sînt afişate în partea de jos a ecranului.

ii) Calculatorul se opreşte şi aşteaptă introducerea unei expresii de la tastatură. Această expresie este apoi atribuită ca valoare variabilei în cauză. Calculatorul semnalează că aşteaptă o intrare prin semnul întrebării. În cazul cînd la tastarea unei intrări se face o greşeală de sintaxă apare în dreptul ei semnul întrebării clipind. Dacă primul caracter care se tastează într-un INPUT este STOP, execuţia programului este oprită şi se afişează eroare H.

Pentru expresii de tip string, bufferul de intrare este iniţializat şi va conţine 2 glilimele afişate şi în partea de jos a ecranului cuprinzînd cursorul care indică modul de lucru (L, C, G, E) în care se aflã calculatorul.

În caz că cele 2 ghilimele sînt şterse se poate apăsa primul caracter STOP şi apoi execuţia programului. Dar ştergînd ghilimelele şi introducînd o expresie (corectă) de tip string este afişat semnul întrebării semnalînd eroare.

iii) Similar cu cazul ii) exceptînd faptul că intrarea este tratată ca un string fără ghilimele iar mecanismul STOP nu funcţionează.

Pentru oprirea programului trebuie tastat CAPS SHIFT cu 6

Exemple: 1. Fie secvenţa de program de mai jos:

```
10 DIM a(5)
20 FOR n=1 TO 5
30 INPUT „Valoarea lui a(n)”; a(n)
40 NEXT n
```

La apelarea acestei secvenţe de program calculatorul va afişa mesajul „Valoarea lui a(n)” după care se aşteaptă introducere valorii numerice pentru a(n). Toată această secvenţă se va repeta de 5 ori, astfel că la terminarea secvenţei vom avea valorile pentru componentele vectorului „a”.

2. Fie secvenţa de program de mai jos:

```
10 INPUT a$
20 PRINT „Eu sînt calculatorul”; a$
30 STOP
```

La execuţia acestei secvenţe de program, calculatorul va aştepta introducerea valorii lui a$ (tip şir) după care va afişa mesajul „Eu sînt calculatorul” şi valoarea lui a$. Dacă a$=„COBRA”, atunci mesajul care va apărea va fi „Eu sînt calculatorul COBRA”.

INVERSE n

Controlează inversarea culorilor pentru caracterele ulterioare. Dacă n=0 caracterele se afişează în video normal. Dacă n=1 afişarea se face în video invers: caractere de culoare paper pe fond ink.

Eroare K dacă n diferit de 0 sau 1.

LET v=e

Valoarea e este atribuită variabilei v. Este obligatorie prezenţa cuvîntului LET. Orice variabilă simplă trebuie definită printr-o instrucţiune de tip LET, READ sau INPUT.

Dacă v este o variabilă de tip şir indexat (string indexat: a$(i)) sau subşir indexat (sliced string) atunci se atribuie lui v o valoare e avînd aceleaşi dimensiuni cu ale variabilei v (şirul e va fi sau truncheat sau umplut la dreapta cu spaţii).

Exemplu: Dacă dorim ca variabila w să aibă valoarea 123, vom folosi instrucţiunea:

LET w=123

LIST n

În partea de sus a ecranului se generează un listing al programului, începînd cu linia n (linia n devine linie curentă, indexată de cursor).

LIST

Listarea programului BASIC, pe ecranul TV, începînd cu linia 0.

LLIST n

Generarea listingului la imprimantă, începînd cu linia n.

LLIST

Generarea listingului la imprimantă, începînd cu linia 0.

LOAD f

(Load File) Încărcarea programului (fişierului) cu numele f şi a variabilelor de la unitatea de memorie externă (caseta magnetică).

LOAD f DATA V()

Încărcarea unei matrici numerice.

LOAD f DATA a$()

Încărcarea unei matrici de stringuri (caractere alfanumerice).

LOAD f CODE m n

Încărcarea a cel mult n ~~biţi~~ octeţi începînd de la adresa m.

LOAD f CODE m

Încărcarea ~~biţilor~~ octeţilor începînd de la adresa m.

LOAD f CODE

Încărcarea ~~biţilor~~ octeţilor înapoi la adresa de la care au fost salvaţi.

LOAD f SCREEN$

LOAD f CODE 16384,6912. Căutarea fişierului corespunzător pe casetă şi încărcarea lui în zona de memorie corespunzătoare ecranului TV.

Exemplele se găsesc la pagina 28.

LPRINT

Similar cu PRINT, dar utilizînd imprimanta.

MERGE f

Similar cu LOAD, fără să fie şterse liniile de program şi variabilele existente în memoria operativă, cu excepţia celor care au acelaşi număr respectiv nume.

MON

Comanda de intrare în programul monitor pentru lucrul în cod maşină. Este o comandă specifică calculatorului COBRA.

Monitorul COBRA are următoarele comenzi:

S adr — Substituţie la adresa adr. Se poate modifica octetul de la adresa respectivă sau se poate afişa octetul anterior (apăsînd SPACE) sau octetul următor (apăsînd ENTER). Se iese tastînd Q.

M adr1 adr2 adr3 — Mută zona de memorie cuprinsă între adresele adr1 şi adr2 la adresa adr3.

F adr lg oc — Umple zona de memorie de la adresa adr, de lungime lg, cu octeţi avînd valoarea oc.

G adr1 [adr2] — Lansează execuţia programului de la adresa adr1 cu oprire opţională la adresa adr2.

D adr — Afişează conţinutul a 128 de locaţii de memorie succesive începînd cu adresa adr.
De asemenea, poate fi afişată pagina de memorie anterioară (tastînd SPACE) sau pagina următoare (tastînd ENTER).

R — Afişarea conţinutului registrelor procesorului.

B — Reîntoarcere la interpretorul BASIC.

MOVE f1 f2

Neutilizabil.

NEW

Restartează sistemul BASIC, după ce şterge programul şi variabilele existente. Memoria este utilizată pînă la (inclusiv) octetul a cărui adresă este conţinută în variabila de sistem RAMBOT. Sînt păstrate valorile variabilelor de sistem UDG, P RAMT, RASP şi PIP.

NEXT a

Instrucţiune utilizată în ciclurile FOR-NEXT.

i) Este căutată variabila de control a.

ii) Se avansează valoarea lui a cu valoarea pasului.

iii) Dacă pasul are o valoare pozitivă şi dacă nu s-a atins limita superioară sau dacă pasul are o valoare negativă şi nu s-a atins limita inferioară, se reia efectuarea ciclului cu prima sa instrucţiune.

Eroare 2 dacă nu există variabila a.

Eroare 1 dacă există o altă variabilă cu acelaşi nume a dar aceasta nu este variabilă de control a ciclului.

OPEN#

Neutilizabil.

OUT m,n

Conţinutul ~~bitului~~ octetului n din memorie este transferat portului m, la nivel de procesor (se încarcă m în perechea de regiştri BC, n în regis-trul A şi se execută instrucţiunea OUT(C), A din limbajul de asamblare Z80). Eroare B dacă m şi n nu sînt în intervalele 0 <= m <= 65535 respectiv –255 <= n <= 255.

OVER n

Controlează supraimprimarea următoarelor caractere astfel:

n=0 noile caractere sînt afişate în locul celor vechi, care sînt şterse

n=1 noile caractere sînt amestecate cu cele anterioare şi rezultă culoarea cernelii (ink), dacă au noul sau vechiul (dar nu ambele) caracter are culoarea cernelii sau rezultă culoarea fondului (paper), dacă şi noul şi vechiul caracter au ambele aceeaşi culoare (a cernelii sau a hîrtiei).

Eroare K dacă n diferit de 0 sau 1.

PAPER n

Similar cu INK dar acţionînd asupra culorii fondului (paper).

PAUSE n

Calculele sînt oprite şi calculatorul face o pauză, cu durata de n/50 secunde sau pînă la apăsarea unei taste.

PLOT c,m,n

Tipăreşte un punct (pixel) avînd coordonatele (|m|, |n|), stabilindu-se aici şi noua poziţie a cursorului grafic. Originea sistemului de coordonate (0,0) pe ecran este stînga-jos.

Dacă valoarea lui c (valoare între 0 şi 7 asociată culorii pentru cerneală (INK); vezi pag. 9) nu se specifică, punctul afişat va fi de culoarea cernelii curente, iar celelalte caracteristici de culoare (culoarea fondului, clipirea şi strălucirea) rămîn neschimbate.

Eroare B dacă m şi n nu sînt în intervalele 0 <= m <= 255 respectiv 0 <= n <= 175.

POKE m,n

În ~~bitul~~ octetul de la adresa m se introduce valoarea n.

Eroare B dacă m şi n nu sînt în intervalele 0 <= m <= 65535 respectiv –255 <= n <= 255.

Exemplu: Dacă la adresa 32568 vrem să avem valoarea 175, vom scrie instrucţiunea:

```
POKE 32568,175
```

PRINT ”...”

”...” reprezintă o secvenţă de caractere tipăribile, separate între ele prin virgule (,) punct-virgulă (;) sau apostroafe (' accente, nu ghilimele), secvenţă care la execuţia instrucţiunii va fi afişată pe ecranul monitorului TV.

Punct-virgula între două caractere ; nu are nici un efect la afişare. Caracterele sînt doar separate între ele.

Virgula , produce apariţia unui caracter de control de tip TAB (caracterul următor se afişează pe aceeaşi linie, la următoarea poziţie de tabulare). (Vezi pag. 10).

Apostroful ' produce introducerea unui caracter ENTER la afişare (CR plus LF (Carriage Return plus Line Feed)).

Dacă instrucţiunea PRINT nu se termină cu punct-virgulă, virgulă sau apostrof, după afişarea şirului de caractere se execută un ENTER (CR+LF).

Un caracter tipăribil (PRINT item) poate fi:

i) Stringul gol (două ghilimele care nu cuprind nimic: ””).

ii) O expresie numerică.

Dacă numărul este negativ se afişează un minus în faţă.

Fie x modulul numărului de afişat.

Dacă x <= 1*E-5 sau x >= 1*E+13 afişarea se va face utilizînd notaţia exponenţială. Mantisa poate avea maxim 8 cifre (fără zerouri în partea dreaptă), cu punctul zecimal pus după prima cifră (fără punct zecimal dacă mantisa are o singură cifră. Urmează un E (semnificînd 10) ridicat la o putere pozitivă sau negativă formată din maxim 2 cifre.

Dacă x este în intervalul 1*E-5 < x < 1*E+13 el va fi afişat în notaţia zecimală obişnuită, cu maxim 8 cifre semnificative şi fără zerouri după punctul zecimal.

Zero este afişat ca o singură cifră 0.

iii) Şir de caractere (string expression).

Simbolurile (cu nume formate din unul sau mai multe caractere alfanumerice) sînt însoţite de caracterul pauză, înainte sau după.

Caracterele de control produc efectul lor de control.

Caracterele ce nu sînt recunoscute sînt înlocuite la afişare prin semnul întrebării ?.

iv) PRINT AT m,n; „ ... ”

Mesajul „ ... ” este afişat pe ecran începînd din linia m, coloana n.

[0 <= m <= 21 şi 0 <= n <= 31].

v) PRINT TAB n; „ ... ”

Mesajul „ ... ” este afişat cu începere din coloana n. Atenţie, numărul coloanei se obţine împăprţind pe n la 32 şi păstrînd restul (n redus modulo 32). Astfel TAB 33 este identic cu TAB 1.

vi) Caracteristica de culoare, de forma PAPER, INK, FLASH, BRIGHT, INVERSE, OVER se poate specifica înaintea listei de variabile ce trebuie tipărite.

RANDOMIZE n

Stabileşte variabila de sistem (numită SEED), utilizată la generarea următorii valori RND. Dacă n = 0 lui SEED i se atribuie valoarea unei alte variabile de sistem (numită FRAMES) ce contorizează cadrele ce apar pe monitorul TV (şi este astfel un număr aleator).

RANDOMIZE este notat pe tastatură RAND.

Eroare B dacă n nu este în intervalul [0,65535].

RANDOMIZE

Echivalent cu RANDOMIZE 0.

READ v1, v2, ..., vK

Variabilelor v1, v2, ..., vK li se atribuie valorile succesive din lista de valori DATA. RESTORE n forţează citirea datelor începînd cu linia n.

Eroare C pentru tipuri greşite de expresii.
Eroare E dacă lista variabilelor este mai mare ca cea a datelor.
Vezi exemplul de la instrucţiunea DATA.

REM ...

Linie de comentariu. Toate caracterele între această instrucţiune şi ENTER nu sînt interpretate. Deci linia de comentariu nu poate cuprinde şi altă instrucţiune chiar dacă este separată prin 2 puncte.

RESTORE n

Obligă ca citirea datelor printr-o secvenţă READ DATA să se facă începînd cu linia n din tabela de date.

RESTORE

Echivalent cu RESTORE 0.

RETURN

Revenire la prima linie după cea care conţine instrucţiunea de salt GO SUB, pentru continuarea programului, după execuţia unei subrutine. Instrucţiune de încheiere a unei subrutine.
Eroare 7 dacă în stivă nu se află adresa de revenire, din pricina unei erori de programare.

RUN n

Se execută un CLEAR urmat de un GO TO n (start pentru rularea programului, începînd cu linia n).

RUN

Echivalent cu RUN 0.

SAVE f

(Save File f) Comanda de salvare a programului şi a variabilelor în memoria externă (înregistrare pe caseta magnetică).

SAVE f LINE m

Salvarea programului astfel încît la reîncărcarea lui în memoria operativă se execută şi un salt automat la linia m.

SAVE f DATA v()

Salvarea unei matrici numerice.

SAVE f DATA a$()

Salvarea unei matrici de stringuri.

SAVE f CODE m,n

Salvarea a n ~~biţi~~ octeţi începînd de la adresa m.

SAVE f SCREEN$

SAVE f CODE 16384,6912. Salvarea zonei de memorie corespunzătoare ecranului TV în fişierul f, pe casetă.

Eroare F dacă f nu cuprinde nici un caracter sau dacă f are lungimea mai mare de 10 caractere.

Exemple de folosire a instrucţiunilor LOAD şi SAVE:

1) Fie programul BASIC cu numele „TESTE”. Acest program se va salva şi apoi se va putea reciti de pe banda magnetică folosind instrucţiunile:

   SAVE ”TESTE” şi **LOAD ”TESTE”**

2) Dacă programul „PROG” a fost scris în cod de Z-80, el se va salva şi reciti cu instrucţiunile:

   SAVE ”PROG” CODE şi **LOAD ”PROG” CODE**

   cu precizarea că trebuie furnizate ca parametri adresa de la care să se salveze programul şi lungimea lui (parametrii m şi n din SAVE f CODE m,n).

3) Dacă folosim instrucţiunile:

   SAVE ”BASIC” LINE 10 şi LOAD ”BASIC”

   se va salva programul cu numele BASIC, apoi la încărcarea lui în memorie se va lansa automat în execuţie de la linia 10.

STOP

Opreşte execuţia unui program cu mesajul de eroare 9. CONTINUE provoacă reluarea programului, începînd cu următoarea instrucţiune.

VERIFY

Similar cu LOAD, doar că datele nu se încarcă de pe casetă în RAM, ci sînt doar comparate cu datele deja existente în RAM. Serveşte la verificarea corectitudinii salvării programelor pe caseta magnetică.

Eroare R dacă în urma comparaţiei se pun în evidenţă diferenţe.

MESAJE DE EROARE

Acestea apar în partea de jos a ecranului TV, în cazul în care calculatorul se opreşte din execuţia unui program, opriri normale sau cauzate de erori.

Mesajele sînt codificate (o cifră sau o literă), sînt însoţite de o scurtă explicaţie şi precizează numărul liniei şi numărul de ordine al instrucţiunii eronate din linia respectivă, care a cauzat întreruperea. O comandă direct executabilă este indicată cu numărul de linie 0. Pe o linie, prima instrucţiune are numărul de ordine 1, instrucţiunea 2 este situată după prima coloană sau după THEN, ş.a.m.d.

Comportarea lui CONTINUE depinde foarte mult de mesajul afişat de calculator. În mod normal CONTINUE produce reluarea de la linia şi instrucţiunea specificate în ultimul mesaj apărut, cu excepţia mesajelor 0, 9 şi D.

În continuare, se prezintă o listă cu mesaje, arătîndu-se şi circumstanţele în care pot ele să apară:

Codul	Semnificaţia	Situaţia
0	OK (Totul este în ordine). Marchează executarea unui program sau a unei comenzi ori un salt la o linie cu numărul mai mare decît al oricărei linii din program.	Oricare
1	NEXT fără FOR. Variabila de control al unui ciclu FOR-NEXT nu a fost definită printr-o instrucţiune FOR dar există o variabilă obişnuită cu acelaşi nume.	NEXT
2	Variabilă ce nu poate fi găsită. În cazul unei variabile simple, atunci cînd se încearcă utilizarea ei înainte ca variabila să fie definită printr-o instrucţiune de tip LET, READ, INPUT, FOR sau înainte ca variabila să fie încărcată de pe casetă. În cazul variabilelor indexate, dacă acestea sînt folosite înainte ca ele să fie dimensionate printr-o instrucţiune DIM sau încărcate de pe bandă.	Oricare
3	Indexare eronată. Indice mai mare ca dimensiunile matricii sau număr eronat de indici. Dacă indicele este negativ sau mai mare ca 65535 rezultă eroare B.	Variabile indexate. Substringuri.
4	Depăşirea memoriei. Memoria operativă insuficientă pentru programul respectiv.	LET, INPUT, FOR, DIM, GOSUB, LOAD, MERGE sau la efectuarea calculelor

Codul	Semnificaţia	Situaţia
5	Depăşirea ecranului. Într-o instrucţiune INPUT s-a încercat generarea a mai mult de 23 de linii în partea de jos a ecranului. De asemenea, la încercarea PRINT AT 22.	INPUT PRINT
6	Număr prea mare. Rezultatul calculelor, mai mare ca 1*E+38.	Calcule aritmetice
7	RETURN folosită fără GO SUB.	RETURN
9	Urmare a executării unei instrucţiuni STOP. CONTINUE produce reluarea programului cu următoarea instrucţiune.	STOP
A	Argumentul unei funcţii este necorespunzător.	SQR, LN, ASN, ACS, USR cu argument string
B	Întreg ce depăşeşte domeniul. Cînd se face rotunjirea unui număr la întregul cel mai apropiat şi acesta depăşeşte un anumit domeniu.	RUN, RANDOMIZE, POKE, DIM, GOTO, GOSUB, LIST, PAUSE, PLOT, CHR$, PEEK, USR
C	Nonsens în BASIC. Textul introdus nu formează o expresie validă.	VAL, VAL$
D	BREAK-CONT repeats. S-a tastat BREAK în decursul operării cu un periferic. CONTINUE se comportă normal, reluarea făcîndu-se cu repetarea instrucţiunii unde s-a produs întreruperea.	LOAD, VERIFY, SAVE, MERGE, LLIST, LPRINT, COPY, sau cînd la întrebarea „scroll?” s-a tastat N, SPACE sau STOP.
E	Date insuficiente. S-a încercat citirea prin READ a mai multor date decît cuprinde lista DATA.	READ
F	Nume eronat de fişier. SAVE fără nume sau cu nume mai lung de 10 caractere.	SAVE
G	Memorie prea puţină pentru a putea cuprinde linia de program ce se introduce.	Introducerea unei noi linii de program.
H	STOP în INPUT. S-a tastat STOP în loc de valori numerice în timpul execuţiei unei instrucţiuni INPUT sau s-a tastat într-o instrucţiuni INPUT LINE. CONTINUE se comportă normal reluînd programul cu instrucţiunea INPUT (inclusiv), în afară de cazul cînd se afişează mesajul 9 (în loc de H).	INPUT

Codul	Semnificaţia	Situaţia
I	FOR fără NEXT. Bucla FOR de executat de 0 ori (de ex. FOR n=1 TO 0). Instrucţiunea NEXT corespunzătoare ciclului nu poate fi găsită.	FOR
J	Dispozitiv de intrare-ieşire greşit.	Operaţii de intrare-ieşire.
K	Caracteristica de culoare greşită. Numărul introdus pentru o caracteristică de culoare este incorect.	INK, PAPER, OVER, BORDER, INVERSE, FLASH, BRIGHT
L	BREAK (întrerupere) într-un program. Tastarea unui BREAK este detectată între execuţia a două instrucţiuni, pe parcursul rulării unui program. Numărul liniei şi al instrucţiunii, care apar în mesaj, se referă la instrucţiunea dinainte ca BREAK să fi fost apăsat; CONTINUE produce reluarea programului cu următoarea instrucţiune (permiţîndu-se efecuarea salturilor). În acest mod nu se repetă nici o instrucţiune.	Oricare
M	RAMTOP incorect. Numărul atribuit variabilei de sistem RAMTOP este sau prea mare sau prea mic.	CLEAR, RUN
N	Pierderea unei instrucţiuni. Salt la o instrucţiune care a fost ştearsă din program.	RETURN, NEXT, CONTINUE
P	FN fără DEF. Apelarea unei funcţii utilizator nedefinită.	FN
Q	Parametru eronat. Număr greşit de argumente sau tip greşit de argument (string în loc de număr sau invers).	FN
R	Eroare la încărcarea unui program de pe banda magnetică. Fişierul căutat pe bandă a fost găsit, dar nu poate fi citit sau nu poate fi verificat.	VERIFY, LOAD, MERGE

4. UTILIZAREA PORTURILOR DE INTRARE/IEŞIRE

Microprocesorul Z80 poate accesa 65536 porturi de intrare/ieşire (input/output=I/O ports), pentru a comunica cu exteriorul (tastatura, interfaţa serială RS232 (imprimanta)). Prin intermediul interpretorului BASIC se pot controla porţiunile de intrare-ieşire, astfel:

– intrările, cu funcţia IN, în forma:

IN adresa

rezultatul fiind valoarea octetului citit din portul a cărui adresă a fost specificată;

– ieşirile, cu declaraţia OUT, în forma:

OUT adresa, valoare

rezultatul fiind înscrierea valorii în portul a cărui adresă a fost specificată.

Una dintre adresele celor 65536 porturi de intrare/ieşire are în binar lungimea de 16 biţi (A=Adresa):

A15, A14, A13, ..., A2, A1, A0.

Biţii A0, A1, A2, A3, A4 au în mod normal valoarea 1.

Se poate atribui valoarea 0 doar unuia dintre aceşti 5 biţi, la un anumit moment.

Biţii A6 şi A7 sînt liberi (pot fi folosiţi de utilizator).

Biţii A8, ..., A15 sînt folosiţi uneori pentru informaţii suplimentare.

O valoare citită din sau înscrisă într-un port de intrare/ieşire are lungimea de 8 biţi (D=Data):

D7, D6, ..., D1, D0.

În continuare sînt prezentate porturile de intrare-ieşire utilizate de calculatorul COBRA.

Tastatura este împărţită în 8 jumătăţi de linie (8 semirînduri) de cîte 5 taste fiecare:

FEFE	0	IN 65278 citeşte tastele CAPS SHIFT pînă la
7FFE	7	IN 32766 citeşte tastele SYMBOL SHIFT pînă la B
FDFE	1	IN 65022 citeşte tastele A pînă la G
BFFE	6	IN 49150 citeşte tastele ENTER pînă la H
FBFE	2	IN 64510 citeşte tastele Q pînă la T
DFFE	5	IN 57342 citeşte tastele P pînă la Y
F7FE	3	IN 63486 citeşte tastele 1 pînă la 5
EFFE	4	IN 61438 citeşte tastele 0 pînă la 6

Valorile acestor adrese sînt:

$$254+256*(255-2^n)$$

cu n variind între 0 şi 7.

Biţii octeţilor citiţi prin funcţiile IN de la adresele de mai sus reprezintă: D0 tasta din exteriorul semirîndului, D1 următoarea tastă; D2, D3, D4 taste din interiorul semirîndului şi au valoarea 0 dacă tasta este apăsată şi valoarea 1 dacă tasta nu este apăsată.

Bitul D5 este asociat cu tastele suplimentare: ESC, LINE FEED, NO SCROLL, CTRL, F1, F2, F3, F4.

Bitul D6 este asociat conectorului de casetofon (intrare de casetofon).

Bitul D7 este asociat intrărilor prin interfaţa serială (imprimanta).

Prin biţii portului de ieşire 254 (OUT 254, valoare) se trimit semnalele:

D0, D1, D2 — culoarea pentru borderul imaginii pe monitorul TV;
D3 — ieşire pentru casetofon (salvare pe casetă);
D4 — difuzorul calculatorului;
D7 — ieşire prin interfaţa serială (imprimanta).

La portul de intrare 223 (IN 223) se poate lega un Joystick compatibil Kempston.

5. MEMORIA

Orice informaţie este memorată de calculator sub formă de numere cu valori între 0 şi 255 (cuvinte de 8 biţi = 1 byte = 1 octet). Fiecare octet al memoriei are o adresă, care este un număr între 0 şi FFFFh (deci o adresă ocupă 2 byte = 16 biţi).

Ne putem deci imagina memoria ca us şir de locaţii (numerotate între 0 şi FFFFh, în fiecare locaţie putînd fi memorat un nuimăr cu valoarea între 0 şi 255.

Calculatorul COBRA are 64 Kocteţi (65536) locaţii de memorie de tip RAM ŞI 16 Kocteţi memorie de tip ROM (EPROM).

Conţinutul locaţiilor ambelor memorii (ROM+RAM) poate fi citit cu funcţia PEEK:

PEEK adresa

şi rezultatul este conţinutul acelei adrese.

Se poate modifica conţinutul locaţiilor (numai în memoria RAM), cu instrucţiunea POKE:

POKE adresa, valoare

rezultatul fiind înscrierea valorii în locaţia de memorie specificată de adresa (adresa trebuie să fie un număr între 0 şi 65535, iar valoarea un număr între 0 şi 255).

În continuare este prezentată harta memoriei:

Adresa	**Conţinutul**
00000 16383	Interpretorul BASIC
16384 22527	Memoria ecran (4000-57FF)
22528 23295	Atributele de culoare (5AFF)
23296 23551	Memoria tampon pentru imprimantă (5B00-5BFF)

23552 23733	Variabile de sistem (5C00-5CB5)
CHAN	Informaţii de canal 80H
PROG	Programe BASIC
VARS	Variabile BASIC 80H
E LINE	Comenzi sau linii de program editate NL 80H
WORKS P	Date de intrare (INPUT data) NL Spaţiu de lucru temporar
STKBOT	Stivă pentru calcule aritmetice
STKEND	Spaţiu liber
SP	Stivă de sistem Stivă GOSUB 7
RAM TOP	3EH
UDG	Caractere grafice definite utilizator
P RAMT	

386E=14446
3CFF=15615 FREE

0066=102
0072=114 NMI

6. VARIABILELE DE SISTEM

Locaţiile de memorie între 23552 şi 23733 sînt rezervate variabilelor de sistem. Valorile lor pot fi citite (cu PEEK) şi unele valori pot fi modificate (cu POKE). În continuare este prezentată o listă a acestor variabile.

Abrevierile din prima coloană au semnificaţia:

X = modificarea valorii variabilei poate deranja (puternic) funcţionarea sistemului;

N = valoarea poate fi modificată;

n = (număr) = numărul de octeţi ocupat de variabilă.

La variabilele de doi octeţi, primul octet este cel mai puţin semnificativ. Pentru introducerea valorii V, variabilei pe doi octeţi de la adresa n, se poate folosi secvenţa:

POKE n, V-256*INT(V/256);
POKE n+1, INT(V/256);

iar citirea valorii se face cu:

PEEK n+256*PEEK(n+1)

	Adresa	Nume	Conţinut
N8	23552	KSTATE	Folosită în citirea tastaturii
N1	23560	LAST K	Memorează ultima tastă apăsată
1	23561	REPDEL	Durata (în 1/50 sec.) cît trebuie ţinută o tastă apăsată pentru a se repeta; iniţial 35
1	23562	REPPER	Perioada (în 1/50 sec.) de repetiţie a unei taste menţinută apăsată; valoarea iniţială 5
N2	23563	DEFADD	Adresa argumentelor funcţiilor definite utilizator; iniţial 0
N1	23565	K DATA	Al doilea octet pentru controlul culorii introdus de la tastatură

	Adresa	Nume	Conţinut
N2	23566	TVDATA	Controlul culorii, al lui AT şi TAB pentru TV
X18	23568	STRMS	Adresa canalului ataşat căii
1	23600	NRCOLMX	Numărul maxim de coloane la imprimantă. Valoarea iniţială 32.
1	23601	NRLINES	Numărul de linii care vor fi scrise pe pagina nouă.
1	23602	COPY AS	Adresa zonei de serializare COPY−SCREEN. Valoarea iniţială 60H
1	23603	COPY AA	Adresa zonei de atribute de culoare COPY−SCREEN. Valoarea iniţială 53H
2	23604	BAUD	Baud Rate RS−232. Viteza de transmitere a caracterelor către imprimantă. Valoarea iniţială 006EH 300 . . . . . 01C1H 600 . . . . . 00E4H 1200 . . . . . 006EH 2100 . . . . . 0035H 3800 . . . . . 0019H 9600 . . . . . 000BH 19200 . . . . . 0005H
2	23606	CHARS	Adresa generatorului de caractere minus 256
1	23608	RAPS	Durata sunetului de eroare
1	23609	PIP	Durata sunetului la apăsarea unei taste
1	23610	ERR NR	Codul de mesaje minus 1
X1	23611	FLAGS	Diferiţi indicatori de control ai sistemului BASIC
X1	23612	TVFLAG	Indicatori asociaţi cu televizorul
X2	23613	ERR SP	Adresa elementului din stivă utilizat ca adresă de întoarcere în caz de eroare.
N2	23615	LIST SP	Adresa de întoarcere la listările automate
N1	23617	MODE	Specifică cursorul (K, L, C, E, G)
2	23618	NEWPPC	Linia la care se sare
1	23620	NSPPC	Numărul instrucţiunii în linia la care se sare
2	23621	PPC	Numărul liniei pentru instrucţiunea în execuţie

	Adresa	Nume	Conţinut
1	23623	SUBPPC	Numărul instrucţiunii din linie în execuţie
1	23624	BORDCR	Culoarea border-ului
2	23625	E PPC	Numărul liniei curente
X2	23627	VARS	Adresa variabilelor
N2	23629	DEST	Adresa variabilelor asignate
X2	23631	CHANS	Adresa datelor de canal
X2	23633	CURCHL	Adresa informaţiei curente folosită pentru intrare sau ieşire
X2	23635	PROG	Adresa programului BASIC
X2	23637	NXTLIN	Adresa următoarei linii din program
X2	23639	DATADD	Adresa ultimului element din lista DATA
X2	23641	E LINE	Adresa comenzii introduse
2	23643	K CUR	Adresa cursorului
X2	23645	CH ADD	Adresa următorului caracter care urmează să fie interpretat
2	23647	XPTR	Adresa caracterului după semnul întrebării
X2	23649	WORKSP	Adresa spaţiului de lucru temporar
X2	23651	STKBOT	Adresa inferioară a stivei calculator
X2	23653	STKEND	Adresa de început a spaţiului liber
N1	23655	BREG	Registrul B al calculatorului
N2	23656	MEM	Adresa spaţiului folosit pentru memoria calculatorului
1	23658	FLAGS2	Alţi indicatori
X1	23659	DF SZ	Numărul liniilor din partea de jos a ecranului
2	23660	S TOP	Numărul liniei de sus a programului la listarea automată
2	23662	OLDPPC	Numărul liniei la care sare CONTINUE
1	23664	OSPCC	Numărul de linie la care sare CONTINUE

	Adresa	Nume	Conţinut
N1	23665	FLAGX	Diverşi indicatori
N2	23666	STRLEN	Lungimea asignată şirului
N2	23668	T ADDR	Adresa următorului element din tabela sintaxă
2	23670	SEED	Variabila pentru RND
3	23672	FRAMES	Contorul de cadre
2	23675	UDG	Adresa primului grafic definit de utilizator
1	23677	COORDS	Coordonata x a ultimului punct plotat
1	23678		Coordonata y a ultimului punct plotat
1	23679	P POSN	Numărul poziţiei de scriere pe ecran
1	23680	PR CC	Octetul mai puţin semnificativ al adresei pentru noua poziţie la care se imprimă prin LPRINT
1	23681		Nefolosit
2	23682	ECHO E	Numărul coloanei şi al liniei
2	23684	DF CC	Adresa de afişare pe ecran prin PRINT
2	23686	DFCCL	Acelaşi lucru pentru partea de jos a ecranului
X1	23688	S POSN	Numărul coloanei pentru PRINT
X1	23689		Numărul liniei pentru PRINT
X2	23690	SPOSNL	Ca S POSN pentru partea de jos a ecranului
1	23692	SCR CT	Numără defilările de ecran
1	23693	ATTR P	Culoarea curentă
1	23694	MASK P	Folosit pentru culori transparente
N1	23695	ATTR T	Culori temporare
N1	23696	MASK T	Ca MASK P dar temporar
1	23697	PFLAG	Alţi indicatori

	Adresa	Nume	Conţinut
N30	23698	MEMBOT	Arie memorie calculator
2	23728		Nefolosit
2	23730	RAMTOP	Adresa ultimului octet din aria sistemului BASIC
2	23732	P-RAMT	Adresa ultimului octet de RAM

7. UTILIZAREA CODULUI MAŞINĂ

Acest paragraf se adresează celor care cunosc setul de instrucţiuni Z80 de programare în cod maşină (limbaj de asamblare).

Introducerea programului scris în cod maşină se face în general cu specificarea adresei de început (cel mai bine este ca această adresă să se afle între zona BASIC şi zona caracterelor grafice definite de utilizator).

La pornire, limita superioară a memoriei RAM (RAMTOP) se află la adresa 65366:

```
00000
|
|
65366=RAMTOP
65367=UDG
|
|       =Caractere grafice definite de utilizator
|
65534=P RAMT
```

Se poate deplasa RAMTOP, rezervîndu-se un spaţiu de 100 de octeţi pentru cod maşină începînd cu adresa 65267, cu comanda

CLEAR 65266

```
|
|
65266=RAMTOP
|
|       =100 de octeţi liberi
|
65367=UDG
|
|       =Caractere grafice definite de utilizator
|
65534=P RAMT
```

Introducerea unui program în cod maşină se poate face prin intermediul unui program BASIC, de genul:

```
10    LET a=32500
20    READ n: POKE a,n
30    LET a=a+1: GO TO 20
40    DATA 1, 99, 0, 201
```

Aici s-a introdus programul:

```
LD BC, 99
RET
```

transpus în cod maşină ca:

1, 99, 0 (pentru **LD BC, 99**) şi **201** (pentru **RET**).

Rularea programului BASIC de mai sus, introduce valorile din linia 40, începînd de la adresa 32500.

Execuţia unui program în cod maşină este comandată prin funcţia USR cu argument numeric, reprezentînd adresa primului octet al programului:

PRINT USR adresa de început

După execuţia programului în cod se tipăreşte valoarea conţinută în perechea de registre BC.

În exemplul de mai sus,

PRINT USR 32500

produce apariţia valorii 99.

Reîntoarcerea din cod Z80 în BASIC se face cu instrucţiunea RET.

În rutinele scrise în cod maşină nu se pot utiliza registrele IY şi HL.

Un program în cod maşină poate fi salvat pe caseta magnetică cu:

SAVE ”nume” CODE adr, n

unde **n** reprezintă lungimea programului (număr de octeţi) şi **adr** este adresa de început.

Un program în cod maşină nu se poate autolansa în execuţie după încărcarea de pe casetă.

El poate fi lansat însă de un program ”nume 1”, în BASIC:

```
10    LOAD ”nume” CODE adr, n
20    PRINT USR adr
```

salvat la rîndul său (cu autostartare), prin:

SAVE ”nume 1” LINE 10

Pentru execuţie se încarcă programul ”nume 1” ajutător:

LOAD ”nume 1”

care se autostartează, încarcă programul ”nume”, în cod maşină (linia 10) şi apoi lansează în execuţie programul ”nume” în cod maşină (linia 20).

8. SETUL DE CARACTERE

În continuare este prezentat setul de caractere al calculatorului COBRA, cu codurile în zecimal şi hex. Coloanele din dreapta dau mnemonicele corespunzătoare în limbaj de asamblare. Anumite instrucţiuni Z80 sînt compuse, începînd cu CB sau ED.

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
0	—	00	NOP	RLC B	
1	—	01	LD BC, nn	RLC C	
2	—	02	LD (BC), A	RLC D	
3	— nefolosite:	03	INC BC	RLC E	
4	—	04	INC B	RLC H	
5	—	05	DEC B	RLC L	
6	PRINT virgulă	06	LD B, n	RLC (HL)	
7	EDIT	07	RLCA	RLC A	
8	cursor stînga	08	EX AF, AF'	RRC B	
9	cursor dreapta	09	ADD HL, BC	RRC C	
10	cursor jos	0A	LD A, (BC)	RRC D	
11	cursor sus	0B	DEC BC	RRC E	
12	DELETE	0C	INC C	RRC H	
13	ENTER	0D	DEC C	RRC L	
14	număr	0E	LD C, n	RRC (HL)	
15	nefolosit	0F	RRCA	RRC A	
16	INK control	10	DJNZ dis	RL B	
17	PAPER control	11	LD DE, nn	RL C	
18	FLASH control	12	LD (DE), A	RL D	
19	BRIGHT control	13	INC DE	RL E	
20	INVERSE control	14	INC D	RL H	
21	OVER control	15	DEC D	RL L	
22	AT control	16	LD D, n	RL (HL)	
23	TAB control	17	RL A	RL A	
24	—	18	JR dis	RR B	
25	—	19	ADD HL, DE	RR C	
26	—	1A	LD A, (DE)	RR D	
27	— nefolosite	1B	DEC DE	RR E	
28	—	1C	INC E	RR H	
29	—	1D	DEC E	RR L	
30	—	1E	LDE n	RR (HL)	
31	—	1F	RR A	RR A	
32	Spaţiu	20	JR NZ, dis	SLA B	
33	!	21	LD HL, nn	SLA C	
34	”	22	LD (nn), HL	SLA D	
35	#	23	INC HL	SLA E	
36	$	24	INC H	SLA H	
37	%	25	DEC H	SLA L	

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
38	&	26	LD H, n	SLA (HL)	
39	'	27	DAA	SLA A	
40	(	28	JR Z, dis	SRA B	
41	)	29	ADD HL, HL	SRA C	
42	*	2A	LD HL, (nn)	SRA D	
43	+	2B	DEC HL	SRA E	
44	,	2C	INC L	SRA H	
45	–	2D	DEC L	SRA L	
46	.	2E	LD L, n	SRA (HL)	
47	/	2F	CPL	SRA A	
48	0	30	JR NC, dis		
49	1	31	LD SP, nn		
50	2	32	LD (nn), A		
51	3	33	INC SP		
52	4	34	INC (HL)		
53	5	35	DEC (HL)		
54	6	36	LD (HL), n		
55	7	37	SCF		
56	8	38	JR C, dis	SRL B	
57	9	39	ADD HL, SP	SRL C	
58	:	3A	LD A, (nn)	SRL D	
59	;	3B	DEC SP	SRL E	
60	<	3C	INC A	SRL H	
61	=	3D	DEC A	SRL L	
62	>	3E	LD A, n	SRL (HL)	
63	?	3F	CCF	SRL A	
64	∂	40	LD B, B	BIT 0, B	INB, (C)
65	A	41	LD B, C	BIT 0, C	OUT (C), B
66	B	42	LD B, D	BIT 0, D	SBC HL, BC
67	C	43	LD B, E	BIT 0, E	LD (nn), BC
68	D	44	LD B, H	BIT 0, H	NEG
69	E	45	LD B, L	BIT 0, L	RETN
70	F	46	LD B, (HL)	BIT 0, (HL)	IM 0
71	G	47	LD B, A	BIT 0, A	LD I, A
72	H	48	LD C, B	BIT 1, B	IN (C), C
73	I	49	LD C, C	BIT 1, C	OUT (C), C
74	J	4A	LD C, D	BIT 1, D	ADC HL, BC
75	K	4B	LD C, E	BIT 1, E	LD BC, (nn)
76	L	4C	LD C, H	BIT 1, H	
77	M	4D	LD C, L	BIT 1, L	RETI
78	N	4E	LD C, (HL)	BIT 1, (HL)	
79	O	4F	LD C, A	BIT 1, A	LD R, A
80	P	50	LD D, B	BIT 2, B	IN D, (C)
81	Q	51	LD D, C	BIT 2, C	OUT (C), D
82	R	52	LD D, D	BIT 2, D	SBC HL, DE
83	S	53	LD D, E	BIT 2, E	LD (nn), DE

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
84	T	54	LD D, H	BIT 2, H	
85	U	55	LD D, L	BIT 2, L	
86	V	56	LD D, (HL)	BIT 2, (HL)	IM 1
87	W	57	LD D, A	BIT 2, A	LD A, I
88	X	58	LD E, B	BIT 3, B	IN E, (C)
89	Y	59	LD E, C	BIT 3, C	OUT (C), E
90	Z	5A	LD E, D	BIT 3, D	ADC HL, DE
91	[	5B	LD E, E	BIT 3, E	LD DE, (nn)
92	/	5C	LD E, H	BIT 3, H	
93	]	5D	LD E, L	BIT 3, L	
94	^	5E	LD E, (HL)	BIT 3, (HL)	IM 2
95	_	5F	LD E, A	BIT 3, A	LD A, R
96	`	60	LD H, B	BIT 4, B	IN H, (C)
97	a	61	LD H, C	BIT 4, C	OUT (C), H
98	b	62	LD H, D	BIT 4, D	SBC HL, HL
99	c	63	LD H, E	BIT 4, E	LD (nn), HL
100	d	64	LD H, H	BIT 4, H	
101	e	65	LD H, L	BIT 4, L	
102	f	66	LD H, (HL)	BIT 4, (HL)	
103	g	67	LD H, A	BIT 4, A	RRD
104	h	68	LD L, B	BIT 5, B	IN L, (C)
105	i	69	LD L, C	BIT 5, C	OUT (C), L
106	j	6A	LD L, D	BIT 5, D	ADC HL, HL
107	k	6B	LD L, E	BIT 5, E	LD HL, (nn)
108	l	6C	LD L, H	BIT 5, H	
109	m	6D	LD L, L	BIT 5, L	
110	n	6E	LD L, (HL)	BIT 5, (HL)	
111	o	6F	LD L, A	BIT 5, A	RLD
112	p	70	LD (HL), B	BIT 6, B	IN F, (C)
113	q	71	LD (HL), C	BIT 6, C	
114	r	72	LD (HL), D	BIT 6, D	SBC HL, SP
115	s	73	LD (HL), E	BIT 6, E	LD (nn), SP
116	t	74	LD (HL), H	BIT 6, H	
117	u	75	LD (HL), L	BIT 6, L	
118	v	76	HALT	BIT 6, (HL)	
119	w	77	LD (HL), A	BIT 6, A	
120	x	78	LD A, B	BIT 7, B	IN A, (C)
121	y	79	LD A, C	BIT 7, C	OUT (C), A
122	z	7A	LD A, D	BIT 7, D	ACD HL, SP
123	{	7B	LD A, E	BIT 7, E	LD SP, (nn)
124	|	7C	LD A, H	BIT 7, H	
125	}	7D	LD A, L	BIT 7, L	
126	“	7E	LD A, (HL)	BIT 7, (HL)	
127	©	7F	LD A, A	BIT 7, A	
128		80	ADD A, B	RES 0, B	
129		81	ADD A, C	RES 0, C	

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
130		82	ADD A, D	RES 0, D	
131		83	ADD A, E	RES 0, E	
132		84	ADD A, H	RES 0, H	
133		85	ADD A, L	RES 0, L	
134		86	ADD A, (HL)	RES 0, (HL)	
135		87	ADD A, A	RES 0, A	
136		88	ADC A, B	RES 1, B	
137		89	ADC A, C	RES 1, C	
138		8A	ADC A, D	RES 1, D	
139		8B	ADC A, E	RES 1, E	
140		8C	ADC A, H	RES 1, H	
141		8D	ADC A, L	RES 1, L	
142		8E	ADC A, (HL)	RES 1, (HL)	
143		8F	ADC A, A	RES 1, A	
144	(a) |	90	SUB B	RES 2, B	
145	(b) |	91	SUB C	RES 2, C	
146	(c) |	92	SUB D	RES 2, D	
147	(d) |	93	SUB E	RES 2, E	
148	(e) |	94	SUB H	RES 2, H	
149	(f) |	95	SUB L	RES 2, L	
150	(g) |	96	SUB (HL)	RES 2, (HL)	
151	(h) |	97	SUB A	RES 2, A	
152	(i) |	98	SBC A, B	RES 3, B	
153	(j) caractere	99	SBC A, C	RES 3, C	
154	(k) grafice	9A	SBC A, D	RES 3, D	
155	(l) |	9B	SBC A, E	RES 3, E	
156	(m) |	9C	SBC A, H	RES 3, H	
157	(n) |	9D	SBC A, L	RES 3, L	
158	(o) |	9E	SBC A, (HL)	RES 3, (HL)	
159	(p) |	9F	SBC A, A	RES 3, A	
160	(q) |	A0	AND B	RES 4, B	LDI
161	(r) |	A1	AND C	RES 4, C	CPI
162	(s) |	A2	AND D	RES 4, D	INI
163	(t) |	A3	AND E	RES 4, E	OUTI
164	(u) |	A4	AND H	RES 4, H	
165	RND	A5	AND L	RES 4, L	
166	INKEY$	A6	AND (HL)	RES 4, (HL)	
167	PI	A7	AND A	RES 4, A	
168	FN	A8	XOR B	RES 5, B	LDD
169	POINT	A9	XOR C	RES 5, C	CPD
170	SCREEN$	AA	XOR D	RES 5, D	IND
171	ATTR	AB	XOR E	RES 5, E	OUTD
172	AT	AC	XOR H	RES 5, H	
173	TAB	AD	XOR L	RES 5, L	
174	VAL$	AE	XOR (HL)	RES 5, (HL)	
175	CODE	AF	XOR A	RES 5, A	

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
176	VAL	B0	OR B	RES 6, B	LDIR
177	LEN	B1	OR C	RES 6, C	CPIR
178	SIN	B2	OR D	RES 6, D	INIR
179	COS	B3	OR E	RES 6, E	OTIR
180	TAN	B4	OR H	RES 6, H	
181	ASN	B5	OR L	RES 6, L	
182	ACS	B6	OR (HL)	RES 6, (HL)	
183	ATN	B7	OR A	RES 6, A	
184	LN	B8	CP B	RES 7, B	LDDR
185	EXP	B9	CP C	RES 7, C	CPDR
186	INT	BA	CP D	RES 7, D	INDR
187	SQR	BB	CP E	RES 7, E	OTDR
188	SGN	BC	CP H	RES 7, H	
189	ABS	BD	CP L	RES 7, L	
190	PEEK	BE	CP (HL)	RES 7, (HL)	
191	IN	BF	CP A	RES 7, A	
192	USR	C0	RET NZ	SET 0, B	
193	STR$	C1	POP BC	SET 0, C	
194	CHR$	C2	JP NZ, nn	SET 0, D	
195	NOT	C3	JP nn	SET 0, E	
196	BIN	C4	CALL NZ, nn	SET 0, H	
197	OR	C5	PUSH BC	SET 0, L	
198	AND	C6	ADD A, n	SET 0, (HL)	
199	<=	C7	RST 0	SET 0, A	
200	>=	C8	RET Z	SET 1, B	
201	<>	C9	RET	SET 1, C	
202	LINE	CA	JP Z, nn	SET 1, D	
203	THEN	CB	operaţii pe biţi	SET 1, E	
204	TO	CC	CALL Z, nn	SET 1, H	
205	STEP	CD	CALL nn	SET 1, L	
206	DEF FN	CE	ADC A, n	SET 1, (HL)	
207	CAT	CF	RST 8	SET 1, A	
208	FORMAT	D0	RET NC	SET 2, B	
209	MOVE	D1	POP DE	SET 2, C	
210	ERASE	D2	JP NC, nn	SET 2, D	
211	OPEN#	D3	OUT (n), A	SET 2, E	
212	CLOSE#	D4	CALL NC, nn	SET 2, H	
213	MERGE	D5	PUSH DE	SET 2, L	
214	VERIFY	D6	SUB n	SET 2, (HL)	
215	BEEP	D7	RST 10	SET 2, A	
216	CIRCLE	D8	RET C	SET 3, B	
217	INK	D9	EXX	SET 3, C	
218	PAPER	DA	JP C, nn	SET 3, D	
219	FLASH	DB	IN A, (nn)	SET 3, E	
220	BRIGHT	DC	CALL C, nn	SET 3, H	

Cod	Caracterul	Hex Z80	L. asamblare	-după CB	-după ED
221	INVERSE	DD	instrucţiuni care folosesc IX	SET 3, L	
222	OVER	DE	SBC A, n	SET 3, (HL)	
223	OUT	DF	RST 18	SET 3, A	
224	LPRINT	E0	RET PO	SET 4, B	
225	LLIST	E1	POP HL	SET 4, C	
226	STOP	E2	JP PO, nn	SET 4, D	
227	READ	E3	EX (SP), HL	SET 4, E	
228	DATA	E4	CALL PO, nn	SET 4, H	
229	RESTORE	E5	PUSH HL	SET 4, L	
230	NEW	E6	AND n	SET 4, (HL)	
231	BORDER	E7	RST 20	SET 4, A	
232	CONTINUE	E8	RET PE	SET 5, B	
233	DIN	E9	JP (HL)	SET 5, C	
234	REM	EA	JP PE, nn	SET 5, D	
235	FOR	EB	EX DE, HL	SET 5, E	
236	GO TO	EC	CALL PE, nn	SET 5, H	
237	GO SUB	ED	instrucţiuni OUT	SET 5, L	
238	INPUT	EE	XOR n	SET 5, (HL)	
239	LOAD	EF	RST 28	SET 5, A	
240	LIST	F0	RET P	SET 6, B	
241	LET	F1	POP AF	SET 6, C	
242	PAUSE	F2	JP P, nn	SET 6, D	
243	NEXT	F3	DI	SET 6, E	
244	POKE	F4	CALL P, nn	SET 6, H	
245	PRINT	F5	PUSH AF	SET 6, L	
246	PLOT	F6	OR n	SET 6, (HL)	
247	RUN	F7	RST 30	SET 6, A	
248	SAVE	F8	RET M	SET 7, B	
249	RANDOMIZE	F9	LD P, HL	SET 7, C	
250	IF	FA	JP M, nn	SET 7, D	
251	CLS	FB	EI	SET 7, E	
252	DRAW	FC	CALL M, nn	SET 7, H	
253	CLEAR	FD	instrucţiuni care folosesc IY	SET 7, L	
254	RETURN	FE	CP n	SET 7, (HL)	
255	COPY	FF	RST 38	SET 7, A	